Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 385 (2007) 115–126
Theoretical Computer Science
www.elsevier.com/locate/tcs

Multiple genome rearrangement by swaps and by element duplications

V.Yu. Popov

Department of Mathematics and Mechanics, Ural State University, 620083 Ekaterinburg, Russia

Received 20 December 2004; received in revised form 6 April 2007; accepted 23 May 2007

Communicated by A. Apostolico

Abstract

We consider the swap distance and the element duplication distance. We show that the swap centre permutation problem is **NP**-complete. We show that the element duplication centre problem is **NP**-complete.

Keywords: Multiple sorting; Distance function; **NP**-complete

1. Introduction

The study of genome rearrangements has drawn a lot of attention in recent years. Large amounts of genomic data on various organisms may be obtained via experiments, including sequence homology between genes, restriction maps, and other hybridization techniques (see [1] for details).

In the 1980's, Palmer and colleagues [2–6] gave evidence that many different species have essentially the same set of genes, but their order may differ between species. This suggests that global rearrangement events (such as reversals and transpositions of genome segments) can be used to trace the evolutionary path between genomes. Since such events are far more rare then point mutations, one can track the genome rearrangements through the evolutionary history of the species much further back then regular mutations allow. This is done by comparing gene orders in the studied species and reconstructing the sequences of genome rearrangements that have transformed the ancestral genome species into the contemporary species. When studying more than two species the key problem arising is to reconstruct the phylogenetic tree achieving minimal distance, given only permutations at the leaves (contemporary species). The problem can be studied when the topology of the tree is known, or is not known. When the topology of the tree is restricted to a star, this problem is known as the median problem. Also we can consider the centre problem.

The use of gene order data for finding globally optimal phylogenetic trees is inherently difficult. Not only are some measures of genomic distance computationally complex [7,12], but more important, the extension of any of them, even the reversals-distance for signed genomes [8], or the breakpoint distance [9], to three or more genomes

E-mail address: Vladimir.Popov@usu.ru.

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2007.05.029

– multiple genome rearrangement – is **NP**-hard [10,11]. In [13,14] proved that in general case (strings over finite alphabets) determining reversal, transposition or signed reversal distance between two species is **NP**-hard. So, it is interesting to consider some bounded distances. Investigating bounded problems is useful as it gives us a more complete understanding of the computational complexity of these problems in general. Since evolutionary events are far more rare than point mutations, there is a very small chance of reverse mutations that will affect the exact same location on the genome. Furthermore, it is natural to suppose that there are no overlap mutations. In this particular case we can hope to reduce evolutionary events (such as duplications and transpositions) to more simple operations. For example, we can consider

$$ABCD, A = 8\ \ 9\ \ 10, B = 6\ \ 7, C = 3\ \ 4\ \ 5, D = 1\ \ 2,$$

instead

$$8\ \ 9\ \ 10\ \ 6\ \ 7\ \ 3\ \ 4\ \ 5\ \ 1\ \ 2,$$

or

$$ABCBD, A = 1, B = 2\ \ 3\ \ 4, C = 5, D = 6,$$

instead

$$1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 2\ \ 3\ \ 4\ \ 6.$$

In first case we can consider transposition of letters B and C instead transposition of words $6\ \ 7$ and $3\ \ 4\ \ 5$. In last case we can consider duplication of letter B instead duplication of word $2\ \ 3\ \ 4$. So, we can use letters instead words.

In this paper, we consider the swap distance and the element duplication distance.

2. Preliminaries

Given the set $T_n = \{1, 2, \ldots, n\}$, a permutation π of T_n is bijective function $\pi : T_n \to T_n$. The symmetric group S_n is the set of all the permutations of T_n. We can view a permutation $\pi \in S_n$ as an ordered arrangement of the elements in T_n where $\pi[i]$ is the element in position i. In this view, the integers $1, 2, \ldots, n$ are used to indicate both positions and elements. We also can view a permutation $\pi \in S_n$ as

$$\begin{pmatrix} 1 & 2 & \ldots & n \\ \pi[1] & \pi[2] & \ldots & \pi[n] \end{pmatrix}.$$

A swap operation exchanges two elements of a permutation. The swap $S\langle i, j\rangle$, where

$$1 \le i < j \le n,$$

is the permutation

$$\begin{pmatrix} 1 & 2 & \ldots & i-1 & i & i+1 & \ldots & j-1 & j & j+1 & \ldots & n \\ 1 & 2 & \ldots & i-1 & j & i+1 & \ldots & j-1 & i & j+1 & \ldots & n \end{pmatrix}.$$

The minimum number of swaps needed to transform a permutation π_1 into a permutation π_2 (or viceversa) is called the swap distance between π_1 and π_2, here denoted by $d_s(\pi_1, \pi_2)$. Note that a chain of swaps needed to transform a permutation π_1 into a permutation π_2 always exists (see [15], section 5.4.).

Given two permutations, the problem of sorting by swaps calls for finding the swap distance between the two permutations and an associated shortest series of swaps. Sorting by swaps is solvable in polynomial time [16]. For all $X \subseteq S_n$, we denote:

$$C_s(X) \rightleftharpoons \min_{\tau \in S_n} (\max_{\pi \in X} d_s(\tau, \pi)),$$

$$M_s(X) \rightleftharpoons \min_{\tau \in S_n} \left(\sum_{\pi \in X} d_s(\tau, \pi) \right).$$

A swap centre of X is a permutation $\tau \in S_n$ such that

$$\max_{\pi \in X} d_s(\tau, \pi) = C_s(X).$$

A swap median of X is a permutation $\tau \in S_n$ such that

$$\sum_{\pi \in X} d_s(\tau, \pi) = M_s(X).$$

The swap centre permutation problem is the decision problem: given a nonempty $X \subseteq S_n$ and $r \in \mathbb{N}$, is $C_s(X) \leq r$. The swap median permutation problem is the decision problem: given a non empty $X \subseteq S_n$ and $r \in \mathbb{N}$, is $M_s(X) \leq r$.

An element duplication operation copies an element to a new location. Denote by Σ a fixed alphabet $\{a_1, a_2, \ldots, a_p\}$. The element duplication $D\langle i, j\rangle$, where $1 \leq i \leq n$, $1 \leq j \leq n$, is the operation

$$D\langle i, j\rangle(a_{k_1}a_{k_2}\ldots a_{k_n}) = a_{k_1}a_{k_2}\ldots a_{k_j}a_{k_i}a_{k_{j+1}}a_{k_{j+2}}\ldots a_{k_n}.$$

If $j = 0$, then the element duplication $D\langle i, j\rangle$ is the operation

$$D\langle i, 0\rangle(a_{k_1}a_{k_2}\ldots a_{k_n}) = a_{k_i}a_{k_1}a_{k_2}\ldots a_{k_n}.$$

The minimum number of element duplications needed to transform a string w_2 into a string w_1 is called the element duplication distance between w_1 and w_2, here denoted by $d_d(w_1, w_2)$. Note that a chain of element duplications needed to transform a string w_2 into a string w_1 not always exists. In this case $d_d(w_1, w_2) = +\infty$.

For all finite $X \subseteq \Sigma^+$, we denote:

$$C_d(X) \rightleftharpoons \min_{u \in \Sigma^+} (\max_{v \in X} d_d(u, v)),$$

$$M_d(X) \rightleftharpoons \min_{u \in \Sigma^+} \left(\sum_{v \in X} d_d(u, v) \right).$$

Let $occ(y, v)$ denote the number of occurrences of the letter y in the word v. Denote by $c(v)$ a set $\{y \mid y \in \Sigma, occ(y, v) \geq 1\}$. If there exist words $w_1, w_2 \in X$ such that $c(w_1) \neq c(w_2)$, then $d_d(w_1, w_2) = C_d(X) = M_d(X) = +\infty$.

An element duplication centre of X is a string $u \in \Sigma^+$ such that

$$\max_{v \in X} d_d(u, v) = C_d(X).$$

An element duplication median of X is a string $u \in \Sigma^+$ such that

$$\sum_{v \in X} d_d(u, v) = M_d(X).$$

The element duplication centre problem is the decision problem: given a nonempty $X \subset \Sigma^+$ and $r \in \mathbb{N}$, is $C_d(X) \leq r$. The element duplication median problem is the decision problem: given a nonempty $X \subset \Sigma^+$ and $r \in \mathbb{N}$, is $M_d(X) \leq r$.

Similarly, we can consider

$$\min_{u \in \Sigma^+} (\max_{v \in X} d_d(v, u)),$$

$$\min_{u \in \Sigma^+} \left(\sum_{v \in X} d_d(v, u) \right),$$

but in this case we obtane well-studied (see [17–21]) longest common subsequence problem.

3. NP-completeness

3.1. The swap centre permutation problem

Theorem 1. *The swap centre permutation problem is* **NP***-complete.*

Proof. Since sorting by swaps is solvable, it is easy to see that the swap centre permutation problem is in **NP**.

Let us consider the following problem:

CLOSEST STRING PROBLEM:

INSTANCE: A non-negative integer t, positive integers k and L,

$$\{s_1, s_2, \ldots, s_k\} \subseteq \Sigma^L.$$

QUESTION: Is there a string s of length L such that, for all $i = 1, 2, \ldots, k$, $d_H(s, s_i) \leq t$? (Here $d_H(s, s_i)$ denotes the Hamming distance between s and s_i.)

The closest string problem is **NP**-complete even for the restriction to a binary alphabet [22,23]. We will assume that $\Sigma = \{0, 1\}$. Now we transform an instance of the closest string problem into an instance of the swap centre permutation problem as follows:

- $n = 2L$.
- $\pi_i = (2 - s_i[1], 1 + s_i[1], \ldots, 2j - s_i[j], 2j + s_i[j] - 1, \ldots, 2L - s_i[L], 2L + s_i[L] - 1)$ where $s_i = s_i[1]s_i[2]\ldots s_i[L]$.
- $X = \{\pi_i \mid 1 \leq i \leq k\}$.
- $r = t$.

It is easy to see that this transformation can be done in polynomial time and logarithmic space.

Let us show that if there is a string s such that $\max_{1\leq i\leq k} d_H(s, s_i) \leq t$, then $C_s(X) \leq r$. Suppose that $\max_{1\leq i\leq k} d_H(s, s_i) \leq t$, $s = s[1]s[2]\ldots s[L]$, and $\tau = (2 - s[1], 1 + s[1], \ldots, 2j - s[j], 2j + s[j] - 1, \ldots, 2L - s[L], 2L + s[L] - 1)$. By definition

$$\pi_i[2j-1] \neq \tau[2j-1] \Leftrightarrow \pi_i[2j] \neq \tau[2j],$$
$$\pi_i[2j-1] \neq \tau[2j-1] \Leftrightarrow (\pi_i[2j], \pi_i[2j-1]) = (\tau[2j-1], \tau[2j]),$$
$$\pi_i[2j-1] \neq \tau[2j-1] \Leftrightarrow s[j] \neq s_i[j],$$
$$1 \leq i \leq k,$$
$$1 \leq j \leq L.$$

Therefore, for all $i = 1, 2, \ldots, k$,

$$d_s(\tau, \pi_i) \leq d_H(s, s_i).$$

Since $\max_{1\leq i\leq k} d_H(s, s_i) \leq t$, $\max_{\pi_i\in X} d_s(\tau, \pi_i) \leq t = r$. Therefore,

$$C_s(X) \leq r.$$

Now suppose that $C_s(X) \leq r$. Since $C_s(X) \leq r$, there exists a permutation γ such that $\max_{\pi_i\in X} d_s(\gamma, \pi_i) \leq r$. Let

$$N(\pi) = \{l \mid \pi[2l-1] \notin \{2l-1, 2l\}, 1 \leq l \leq L\} \cup$$
$$\{l \mid \pi[2l] \notin \{2l-1, 2l\}, 1 \leq l \leq L\}.$$

Let us prove, by induction on the number $|N(\gamma)|$, that there exists a permutation τ such that

$$\max_{\pi_i\in X} d_s(\tau, \pi_i) \leq \max_{\pi_i\in X} d_s(\gamma, \pi_i)$$

and

$$\tau[2j-1], \tau[2j] \in \{2j-1, 2j\}, 1 \leq j \leq L.$$

Let $|N(\gamma)| = 0$. Then $\tau = \gamma$. This verifies the base of our induction.

Assume that the assertion holds for some p. Let us show that it holds for $p + 1$. Assume that $|N(\gamma)| = p + 1$. Suppose that there exists a positive integer l such that $\gamma[2l-1] \notin \{2l-1, 2l\}$ or $\gamma[2l] \notin \{2l-1, 2l\}$. Fourteen cases are possible:

$$\gamma[2l-1], \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l-1, \gamma[l_2] = 2l, l_1 < l_2 < 2l-1;$$
$$\gamma[2l-1], \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l-1, \gamma[l_2] = 2l, l_2 < l_1 < 2l-1;$$

$\gamma[2l-1], \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l-1, \gamma[l_2] = 2l, 2l < l_1 < l_2;$
$\gamma[2l-1], \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l-1, \gamma[l_2] = 2l, 2l < l_2 < l_1;$
$\gamma[2l-1], \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l-1, \gamma[l_2] = 2l, l_1 < 2l-1, 2l < l_2;$
$\gamma[2l-1], \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l-1, \gamma[l_2] = 2l, l_2 < 2l-1, 2l < l_1;$
$\gamma[2l-1] \notin \{2l-1, 2l\}, \gamma[2l] \in \{2l-1, 2l\}, \gamma[l_1] = 2l-1, 2l < l_1;$
$\gamma[2l-1] \notin \{2l-1, 2l\}, \gamma[2l] \in \{2l-1, 2l\}, \gamma[l_1] = 2l-1, l_1 < 2l-1;$
$\gamma[2l-1] \notin \{2l-1, 2l\}, \gamma[2l] \in \{2l-1, 2l\}, \gamma[l_1] = 2l, 2l < l_1;$
$\gamma[2l-1] \notin \{2l-1, 2l\}, \gamma[2l] \in \{2l-1, 2l\}, \gamma[l_1] = 2l, l_1 < 2l-1;$
$\gamma[2l-1] \in \{2l-1, 2l\}, \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l-1, 2l < l_1;$
$\gamma[2l-1] \in \{2l-1, 2l\}, \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l-1, l_1 < 2l-1;$
$\gamma[2l-1] \in \{2l-1, 2l\}, \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l, 2l < l_1;$
$\gamma[2l-1] \in \{2l-1, 2l\}, \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l, l_1 < 2l-1.$

We consider only the case

$$\gamma[2l-1] \in \{2l-1, 2l\}, \gamma[2l] \notin \{2l-1, 2l\}, \gamma[l_1] = 2l, l_1 < 2l-1$$

because the other thirteen cases can be treated similarly.

Since $\gamma[l_1] = 2l$ and $\gamma[2l] \notin \{2l-1, 2l\}$, $\gamma[2l-1] = 2l-1$. Since $l_1 < 2l-1$, for all $i = 1, 2, \ldots, k$, $\pi_i[l_1] \neq \gamma[l_1]$. By definition, for all $i = 1, 2, \ldots, k$, $\pi_i[2l] \neq \gamma[2l]$. Suppose that, for all $i = 1, 2, \ldots, k$, $d_s(\gamma, \pi_i) = p_i$ and

$$S_i\langle m_{i,1,1}, m_{i,1,2}\rangle, S_i\langle m_{i,2,1}, m_{i,2,2}\rangle, \ldots, S_i\langle m_{i,p_i,1}, m_{i,p_i,2}\rangle$$

is an associated shortest series of swaps. Since $\pi_i[l_1] \neq \gamma[l_1]$, there exists $S_i\langle m_{i,a,1}, m_{i,a,2}\rangle$ such that $m_{i,a,1} = l_1$ or $m_{i,a,2} = l_1$,

$$S_i\langle m_{i,a,1}, m_{i,a,2}\rangle(\pi_i)[l_1] = \gamma[l_1].$$

Since $\pi_i[2l] \neq \gamma[2l]$, there exists $S_i\langle m_{i,b,1}, m_{i,b,2}\rangle$ such that $m_{i,b,1} = 2l$ or $m_{i,b,2} = 2l$,

$$S_i\langle m_{i,b,1}, m_{i,b,2}\rangle(\pi_i)[2l] = \gamma[2l].$$

Therefore, if $\delta = S\langle l_1, 2l\rangle(\gamma)$, then $d_s(\delta, \pi_i) \leq d_s(\gamma, \pi_i)$. It is easy to see that $N(\delta) < N(\gamma)$. Since, for all $i = 1, 2, \ldots, k$, $d_s(\delta, \pi_i) \leq d_s(\gamma, \pi_i)$, then

$$\max_{\pi_i \in X} d_s(\delta, \pi_i) \leq \max_{\pi_i \in X} d_s(\gamma, \pi_i).$$

Since $N(\delta) < N(\gamma)$, by the induction assumption there exists a permutation τ such that

$$\max_{\pi_i \in X} d_s(\tau, \pi_i) \leq \max_{\pi_i \in X} d_s(\delta, \pi_i)$$

and $N(\tau) = 0$. Therefore, there exists a permutation τ such that

$$\max_{\pi_i \in X} d_s(\tau, \pi_i) \leq \max_{\pi_i \in X} d_s(\gamma, \pi_i)$$

and $N(\tau) = 0$.

Consider a permutation τ such that

$$\max_{\pi_i \in X} d_s(\tau, \pi_i) \leq \max_{\pi_i \in X} d_s(\gamma, \pi_i)$$

and $N(\tau) = 0$. Suppose that, for all $i = 1, 2, \ldots, k$, $d_s(\tau, \pi_i) = p_i$ and

$$S_i\langle m_{i,1,1}, m_{i,1,2}\rangle, S_i\langle m_{i,2,1}, m_{i,2,2}\rangle, \ldots, S_i\langle m_{i,p_i,1}, m_{i,p_i,2}\rangle$$

is an associated shortest series of swaps. It is clear that either, for all $j = 1, 2, \ldots, p_i$,

$$S_i\langle m_{i,j,1}, m_{i,j,2}\rangle \in \{S\langle 2l-1, 2l\rangle \mid 1 \le l \le L\}$$

or there exists a permutation β such that

$$\max_{\pi_i \in X} d_s(\beta, \pi_i) < \max_{\pi_i \in X} d_s(\tau, \pi_i)$$

and $N(\beta) = 0$. Therefore, if

$$s = s[1]s[2]\ldots s[L]$$

where

$$s[l] = 1 \Leftrightarrow \tau[2l-1] < \tau[2l],$$

then $\max_{1\le i\le k} d_H(s, s_i) \le t$. Hence we have proved that the swap centre permutation problem is **NP**-complete. □

The minimum number of adjacent element swaps needed to transform a permutation π_1 into a permutation π_2 (or viceversa) is called the adjacent element swap distance between π_1 and π_2, here denoted by $d_{as}(\pi_1, \pi_2)$. Given two permutations, the problem of sorting by adjacent element swaps calls for finding the adjacent element swap distance between the two permutations and an associated shortest series of swaps. Sorting by adjacent element swaps is solvable in polynomial time [16,24]. For all $X \subseteq S_n$, we denote:

$$C_{as}(X) \rightleftharpoons \min_{\tau \in S_n}(\max_{\pi \in X} d_{as}(\tau, \pi)).$$

A adjacent element swap centre of X is a permutation τ such that

$$\max_{\pi \in X} d_{as}(\tau, \pi) = C_{as}(X).$$

The adjacent element swap centre permutation problem is the decision problem: given a nonempty $X \subseteq S_n$ and $r \in \mathbb{N}$, is $C_{as}(X) \le r$.

Corollary 1. *The adjacent element swap centre permutation problem is* **NP***-complete.*

It is clear that the swap centre permutation problem and the adjacent element swap centre permutation problem are **NP**-complete for circular permutations.

3.2. The element duplication centre problem

We need the following problem.

THE SHORTEST COMMON SUPERSEQUENCE PROBLEM (SCS=)

INSTANCE: $\Gamma = \{1, 2, 3\}$, a finite set $S = \{s_1, s_2, \ldots, s_n\}$ of strings $S \subset \Gamma^+$ such that $|s_1| = |s_2| = \cdots = |s_n|$, and a positive integer m.

QUESTION: Is there a string $w \in \Gamma^+$ such that $|w| \le m$ and w is a supersequence of each string $s_i \in S$?

The following theorem is necessary:

Theorem 2. SCS= *is* **NP***-complete.*

Proof. It is easy to see that the SCS= problem is in **NP**.

Let us consider the following problem:

THE SHORTEST COMMON SUPERSEQUENCE PROBLEM (SCS)

INSTANCE: $\Gamma = \{1, 2\}$, a finite set T of strings $T \subseteq \Gamma^+$, and a positive integer k.

QUESTION: Is there a string $u \in \Gamma^+$ such that $|u| \le k$ and u is a supersequence of each string $t_i \in S$?

The SCS problem is known to be **NP**-complete [25]. We will assume that there are n strings $t_1, t_2, \ldots, t_n$ in T.

Let

$$r = \max\{|t_1|, |t_2|, \ldots, |t_n|\} + 1, r_i = r - |t_i|, l = \max\{|r_1|, |r_2|, \ldots, |r_n|\}.$$

Now we transform an instance of the SCS problem to an instance of the SCS= problem as follows:

- $s_i = 3t_i3^{r_i}$.
- $m = k + l + 1$.

It is easy to check that $|3t_13^{r_1}| = |3t_23^{r_2}| = \cdots = |3t_n3^{r_n}|$.

First, suppose that there exists a string $u \in \{1, 2\}^*$ such that $|u| \leq k$ and u is a supersequence of each string $t_i \in T$. Let $w = 3u3^l$. Since $|u| \leq k$, $|3u3^l| \leq k + l + 1 = m$. It is easy to see that if v_1 is a supersequence of each string a_i, $1 \leq i \leq n$, and v_2 is a supersequence of each string b_i, $1 \leq i \leq n$, v_3 is a supersequence of each string c_i, $1 \leq i \leq n$, then $v_1v_2v_3$ is a supersequence of each string $a_ib_ic_i$, $1 \leq i \leq n$. Since $l = \max\{|r_1|, |r_2|, \ldots, |r_n|\}$, 3^l is a supersequence of each string 3^{r_i}, $1 \leq i \leq n$. Therefore, $3u3^l$ is a supersequence of each string $3t_i3^{r_i}$, $1 \leq i \leq n$.

Now suppose that there exists a string $w \in \{1, 2, 3\}^*$ such that $|w| \leq m$ and w is a supersequence of each string $s_i \in S$. Since w is a supersequence of each string $s_i \in S$, and $s_i = 3t_i3^{r_i}$, it is easy to see that $occ(3, w) \geq l + 1$. Let $w = w_13w_23 \ldots w_p3w_{p+1}$ where $w_j \in \{1, 2\}^*$, $1 \leq j \leq p + 1$. Let $w' = 3w_1w_2 \ldots w_pw_{p+1}3^{p-1}$. Since $occ(3, w) \geq l + 1$, $t_i \in \{1, 2\}^*$, and w is a supersequence of each string $s_i \in S$, it is easy to see that w' is a supersequence of each string $s_i \in S$. Since w' is a supersequence of each string $s_i \in S$, and $w_j \in \{1, 2\}^*$, $1 \leq j \leq p + 1$, it is easy to see that $w_1w_2 \ldots w_pw_{p+1}$ is a supersequence of each string $t_i \in T$. In view of $p \geq l + 1$, $m = k + l + 1$, and $|w| \leq m$, it is easy to check that $|w_1w_2 \ldots w_pw_{p+1}| \leq k$. □

Let $\Gamma = \{1, 2, 3\}$, $S = \{s_1, s_2, \ldots, s_n\}$, where $S \subset \Gamma^+$,

$$|s_1| = |s_2| = \cdots = |s_n| = t.$$

We can assume that $occ(i, s_j) \neq 0$, for all $i \in \{1, 2, 3\}$, $j \in \{1, 2, \ldots, n\}$. Let $|w| = m$. If w is a supersequence of each string $s_i \in S$, then it is easy to see that

$$\max_{s_i \in S} d_d(w, s_i) = m - t,$$
$$\sum_{s_i \in S} d_d(w, s_i) = n(m - t).$$

If $C_d(S) = r$ and

$$\max_{s_i \in S} d_d(w, s_i) = C_d(S),$$

then $|w| = r + t$ and w is a supersequence of each string $s_i \in S$. Therefore, the element duplication centre problem is a restricted version of the SCS= problem in which $occ(a, s_i) = 0$ if and only if $occ(a, s_j) = 0$, for all a and $i, j \in \{1, 2, \ldots, n\}$.

Corollary 2. *The element duplication centre problem is* **NP***-hard.*

If $M_d(S) = r$ and

$$\sum_{s_i \in S} d_d(w, s_i) = M_d(S),$$

then $|w| = \frac{r}{n} + t$ and w is a supersequence of each string $s_i \in S$. So the element duplication median problem is a restricted version of the SCS problem in which $occ(a, s_i) = 0$ if and only if $occ(a, s_j) = 0$, for all a and $i, j \in \{1, 2, \ldots, n\}$.

Corollary 3. *The element duplication median problem is* **NP***-hard.*

3.3. Models for tandem arrays

Regularities in a biological sequence can be used to identify the sequence among other sequences, or to infer information about the evolution of the sequence. The genomes of eukaryotes, i.e. higher order organisms such humans, contain many regularities. Tandem repeats, or tandem arrays, which are consecutive occurrences of the same string, are the most frequent.

Traditionally the alignment notation has been used to illustrate a comparison between two or more sequences. Given a set of strings

$$X = \{x_1, x_2, \ldots, x_k\}$$

on an alphabet Σ, a multiple alignment of X is a set of strings

$$A = \{A_1, A_2, \ldots, A_k\},$$

$|A_1| = |A_2| = \cdots |A_k| = n$, on augmented alphabet $\Gamma = \Sigma \cup \{\Delta\}$ such that each string A_i is a copy of x_i into which $n - |x_i|$ copies of special symbol Δ have been inserted. Symbol Δ is called an indel and represents the insertion or deletion of a particular symbol in one string relative to another.

A conventional way to measure the approximate similarity between two sequences $a_1 \ldots a_m$ and $b_1 \ldots b_n$ is to calculate local transformations or costs of local transformations. Usually the considered local transformations are the following:

- substitution: $a_i \to b_j$;
- insertion: $\Delta \to b_j$;
- deletion: $a_i \to \Delta$.

To define a distance between sequences, one should first fix the set of local transformations and nonnegative valued cost function δ that gives for each transformation $a \to b$ a cost $\delta(a, b)$. A penalty matrix specifies the substitution cost for each pair of characters and the insertion/deletion cost for each character. The differences appearing in the considered two sequences can be viewed differently, e.g. one substitution can be viewed as one insertion and one deletion. Therefore, it is natural to observe the minimum number of such differences. The weighted edit distance between x and y is the minimum cost to convert x to y using a penalty matrix.

Tandem arrays are a sequence of repeats that appear adjacent in a string. As concerns biology, such tandemly repeated units are divided into three categories depending on the length of repeated element, the span of the repeated region and its location within the chromosome [26]. Repeats occurring in or near the centromeres and telomeres are called simply satellites. Their span is large, up to a million bases, and the length of the repeated element varies greatly, anywhere from 5 to few hundreds of base pairs.

There are two satellite models (see [27]). One called prefix model and the other consensus model.

A prefix model of a satellite is a string $w \in \Sigma^*$ that approximately matches a train of wagons. A wagon of w is a substring u in string x such that $\delta(w, u) \leq e$. A train of a satellite model w is collection of wagons $u_1, u_2, \ldots, u_p$ ordered by their starting positions in x and satisfying the following properties.

1. $p \geq min\_repeat$, where min_repeat is a fixed parameter that indicates the minimum number of elements a repeating region must contain.

2. $left_{u_{i+1}} - left_{u_i} \in JUMP$, where $left_u$ is the position of the left-end of wagon u in x and

$$JUMP = \{y \mid y \in \cup_{x \in [1, max\_jump]} x \times [min\_range, max\_range]\},$$

with the three parameters min_range, max_range and max_jump fixed.

A prefix model w is said to be valid if there is at least one train of w in the string x. Similarly, a train, when viewed simply as a sequence of substrings of x, is valid if it is the train for some model w.

Consensus model is a prefix model which further satisfies the following property.

3. $left_{u_{i+1}} - right_{u_i} \in GAP$, where $right_u$ is the position of the right-end of wagon u, and

$$GAP = \{y \mid y \in \cup_{x \in [0, max\_jump-1]} x \times [min\_range, max\_range]\}.$$

The distance function in the consensus model is the consensus function, which finds a consensus string of wagons, i.e. string w such that the distance between the string w and each string in $\{u_1, u_2, \ldots, u_p\}$ is at most e. Another way to define a consensus string is to use the consensus error. The consensus error of a string w with respect to a given set $\{u_1, u_2, \ldots, u_p\}$ is the sum of the distances between w and all the strings in $\{u_1, u_2, \ldots, u_p\}$. Parameter $\max\_jump$ allows us to deal with very badly conserved elements inside a satellite (by actually not counting them). Consensus error allows us to deal with relatively badly conserved wagons inside a satellite (and counting them) while we require that the satellite be relatively well conserved globally. This may be useful for compression algorithms for

DNA sequences. It is known that DNA sequences have two characteristic structures. One is reverse complements, and the other is approximate repeats. Compression algorithms for DNA sequences use the structures, and we need to detect as much of this structures as possible.

Let a consensus error model is a string $w \in \Sigma^*$ that approximately matches with consensus error a train of wagons, i.e. $\sum_{i=1}^{p} \delta(w, u_i) \leq e$.

Let us consider the following problem:

THE SATELLITE PROBLEM FOR CONSENSUS ERROR (SPCE):

INSTANCE: Parameters *min_repeat*, *min_range*, *max_range*, *max_jump*, and e, a distance function δ, a string x.

QUESTION: Is there a consensus error model w that is valid for x?

Theorem 3. SPCE *is* **NP**-*complete.*

Proof. It is easy to see that SPCE is in **NP**. Now we transform an instance of the SCS= problem to an instance of SPCE as follows:

- Note that we can assume that $|s_i| \geq 3$. Let $s_i = s_{i,1}s_{i,2}s_{i,3}$ where $s_{i,1}, s_{i,3} \in \{1, 2, 3\}$. Note that we can assume that $s_{i,1} = s_{i,3}$. Let

$$x = s_{1,1}s_{1,2}s_{1,3}s_{2,2}s_{2,3} \ldots s_{n,2}s_{n,3}.$$

- *min_repeat* $= n$
- *min_range* $=$ *max_range* $= |s_1| - 1$
- *max_jump* $= 1$
- Let $L = \{\Delta \to 1, \Delta \to 2, \Delta \to 3\}$ be a set of local transformations, and $\delta(\Delta, 1) = 1$, $\delta(\Delta, 2) = 1$, $\delta(\Delta, 3) = 1$. For convenience, if $\alpha \to \beta \notin L$, then $\delta(\alpha, \beta) = \infty$.
- $e = (m - |s_1|)n$

First suppose that there exists a string $w \in \{1, 2, 3\}^*$ such that $|w| \leq m$ and w is a supersequence of each string $s_i \in S$. Clearly, $\delta(w, s_i) = m - |s_1|$, $1 \leq i \leq n$. Therefore, $\sum_{i=1}^{n} \delta(w, s_i) = (m - |s_1|)n = e$, and w is a consensus error model that is valid for x.

Now suppose that there exists a consensus error model w that is valid for x. In view of *min_repeat* $= n$, *min_range* $=$ *max_range* $= |s_1| - 1$, *max_jump* $= 1$, it is not hard to check that $\sum_{i=1}^{n} \delta(w, s_i) \leq e$. Consider the alignment of $\{s_1, s_2, \ldots, s_n\}$ induced by w: $P_1, P_2, \ldots, P_n$. Let

$$P_i = p_{i,1} \ldots p_{i,k},\ 1 \leq i \leq n,$$
$$w = w_1 \ldots w_k,$$

where

$$p_{i,1}, \ldots, p_{i,k}, w_1, \ldots, w_k \in \{1, 2, 3, \Delta\}.$$

By definition of δ, if $p_{i,j} \neq w_j$, $1 \leq i \leq n$, $1 \leq j \leq k$, then either $p_{i,j} = \Delta$ and $w_j \in \{1, 2, 3\}$ or $\delta(w, s_i) > e$. Therefore, $\delta(w, s_i) = occ(\Delta, P_i) = k - |s_i|$, and w is a supersequence of each string $s_i \in S$. Since

$$e = (m - |s_1|)n,\ \sum_{i=1}^{n} \delta(w, s_i) \leq e,\ \sum_{i=1}^{n} \delta(w, s_i) = (k - |s_i|)n,\ |w| = k,$$

it is easy to see that $|w| \leq m$. □

4. Parameterizations

4.1. The swap centre permutation problem

We consider the following parameterizations of the swap centre permutation problem.

THE SWAP CENTRE PERMUTATION PROBLEM:

INSTANCE: a finite set $X = \{\pi_1, \pi_2, \ldots, \pi_p\}$ of permutations $X \subseteq S_n$ and a positive integer r.

PARAMETERS: p, r.

QUESTION: Is there a permutation τ such that $\max_{\pi \in X} d_s(\tau, \pi) \leq r$?

Theorem 4. *The swap centre permutation problem can be solved in time $O(pn+g(p,r))$ where g is a function which depends only from p and r.*

Proof. Let

$$X = \{\pi_1, \pi_2, \ldots, \pi_p \mid \pi_i \in S_n, i \in \{1, 2, \ldots, p\}\}.$$

Suppose that $\tau \in S_n$ such that

$$\max_{\pi_i \in X} d_s(\tau, \pi_i) = C_s(X).$$

Denote by Y the set of all swaps. Let

$$\alpha\beta = (\alpha[\beta[1]], \alpha[\beta[2]], \ldots, \alpha[\beta[n]])$$

where $\alpha, \beta \in S_n$. Suppose that

$$\pi_i \gamma_{i,1}\gamma_{i,2} \ldots \gamma_{i,d_s(\tau,\pi_i)} = \tau$$

where

$$\gamma_{i,1}, \gamma_{i,2}, \ldots, \gamma_{i,d_s(\tau,\pi_i)} \in Y.$$

For all $\delta \in S_n$, denote by $\gamma_{i,j}^{\delta}$ the permutation $S\langle u, v\rangle$ such that

$$\delta[u], \delta[v] \in \{s, t\},$$
$$\gamma_{i,j} = S\langle s, t\rangle.$$

It is easy to check that

$$\pi_i \gamma_{i,1}\gamma_{i,2} \ldots \gamma_{i,d_s(\tau,\pi_i)} = \tau$$

if and only if

$$\pi_i \delta \gamma_{i,1}^{\delta}\gamma_{i,2}^{\delta} \ldots \gamma_{i,d_s(\tau,\pi_i)}^{\delta} = \tau\delta.$$

Since, for all $1 \leq i < j \leq n$, $S\langle i, j\rangle = (S\langle i, j\rangle)^{-1}$, it is clear that if

$$\max_{\pi_i \in X} d_s(\tau, \pi_i) = C_s(X),$$

then

$$\max_{\pi_i \in X} d_s(\tau\delta, \pi_i\delta) = C_s(\{\pi_i\delta \mid \pi_i \in X\}).$$

Let

$$\delta = (l_1 l_2 \ldots l_m l_{m+1} l_{m+2} \ldots l_n),$$

where

$$\{l_1, l_2, \ldots, l_n\} = \{1, 2, \ldots, n\},$$
$$N = \{l \mid l \in \{1, 2, \ldots, n\}, \pi_i[l] = \pi_j[l], i, j \in \{1, 2, \ldots, p\}\},$$
$$\{l_{m+1}, l_{m+2}, \ldots, l_n\} = N.$$

Let $f|_{\delta}(\pi)$ be a bijective function

$$f|_{\delta} : \{i \mid i \in \{1, 2, \ldots, n\} \backslash N_X\} \to \{1, 2, \ldots, m\},$$

where

$$N_X = \{\pi_1[i] \mid i \in N\}.$$

If $\pi_i[l] = \pi_j[l]$, for some $l \in \{1, 2, \ldots, n\}$ and for all $i, j \in \{1, 2, \ldots, p\}$, then it is easy to see that in this case $\tau[l] = \pi_1[l]$. Therefore,

$$\max_{\pi_i \in X} d_s(\tau, \pi_i) = C_s(X)$$

if and only if

$$\max_{\pi'_i \in X'} d_s(\tau', \pi'_i) = C_s(X),$$

where

$$\{\pi'_1, \pi'_2, \ldots, \pi'_p\} = X',$$
$$\pi'_1, \pi'_2, \ldots, \pi'_p, \tau' \in S_m,$$
$$\{l_1, l_2, \ldots, l_m\} = \{1, 2, \ldots, n\} \backslash N,$$
$$l_1 < l_2 < \cdots < l_m,$$
$$\pi'[s] = f|_\delta(\pi[l_s]),$$
$$\tau'[s] = f|_\delta(\tau[l_s]).$$

So we can suppose that $n \leq m$. It is easy to see that $m \leq 2rp$. Clearly, we can obtain N in time $O(pn)$. Therefore, the swap centre permutation problem can be solved in time $O(pn + g(p, r))$. □

4.2. The element duplication centre problem

We consider the following parameterizations of the element duplication centre problem.

THE ELEMENT DUPLICATION CENTRE PROBLEM:

INSTANCE: $\Sigma = \{a_1, a_2, \ldots, a_p\}$, a finite set $S = \{s_1, s_2, \ldots, s_n\}$ of strings $S \subset \Sigma^+$, and a positive integer r.

PARAMETER: r. (VERSION I)

PARAMETER: n. (VERSION II)

PARAMETER: $L = \max_{s_i \in S} |s_i|$. (VERSION III)

QUESTION: Is there a string w such that $\max_{s_i \in S} d_d(w, s_i) \leq r$?

Similarly, we can define the parameterized version of the element duplication median problem.

Since the element duplication centre problem and the element duplication median problem are restrictions of the SCS problem [28], the element duplication centre problem I and the element duplication median problem I are fixed parameter tractable.

Note that we can use the reduction in the proof of Theorem 2 for unbounded Γ. Since the SCS problem with fixed $|T|$ is W[t]-hard for all t [28], it is easy to see that the element duplication centre problem II and the element duplication median problem II are W[t]-hard for all t.

Let SCS(L) denote the restricted version of SCS in which each input sequence is of length L. It is known that SCS(L) is **NP**-hard for all $L \geq 2$ [29]. In contrast, the element duplication centre problem III can be solved in time $O(1)$. It is interesting from the point of view of DNA sequencing.

Theorem 5. *The element duplication centre problem* III *can be solved in time* $O(1)$ *where* $O(1)$ *is a function which depends only from* L.

Proof. If $\max_{s_i \in S} d_d(w, s_i) \leq r$, then w is a supersequence of each string $s_i \in S$. Therefore,

$$occ(a_i, s_j) = 0 \Leftrightarrow occ(a_i, w) = 0,$$

for all $i \in \{1, 2, \ldots, p\}$, $j \in \{1, 2, \ldots, n\}$. So we can assume that $p \leq L$. Since $\max_{s_i \in S} |s_i| = L$, for all $j \in \{1, 2, \ldots, n\}$, $s_j \in \Sigma^L$. It is clear that $|\Sigma^L| \leq Lp^L$. Since $p \leq L$, it is easy to see that $|\Sigma^L| \leq LL^L$. Therefore, we can suppose that $n \leq LL^L$. Since $p \leq L$ and $n \leq LL^L$, the element duplication centre problem III can be solved in time $O(1)$. □

Similarly, we can prove that the element duplication median problem III can be solved in time $O(1)$.

Note that if we suppose that S is a multiset, then the element duplication centre problem III can be solved in linear time.

References

[1] W.H. Li, D. Graur, Fundamentals of molecular evolution, Sinauer, Sulderland, MA, 1991.
[2] J. Palmer, L. Herbon, Tricircular mitochondrial genomes of Brassica and Raphanus: Reversal of repeat configurations by inversion, Nucleic Acids Research 14 (1986) 9755–9764.
[3] J. Palmer, L. Herbon, Unicircular structure of the Brassica hirta mitochondrial genome, Current Genetics 11 (1987) 565–570.
[4] J. Palmer, L. Herbon, Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence, Journal of Molecular Evolution 28 (1988) 87–89.
[5] J. Palmer, B. Osorio, W. Thompson, Evolutionary significance of inversions in legume chloroplast DNAs, Current Genetics 14 (1988) 65–74.
[6] S.B. Hoot, J.D. Palmer, Structural rearrangements, including parallel inversions, within the chloroplast genome of Anemone and related genera, Journal of Molecular Evolution 38 (1994) 274–281.
[7] A. Caprara, Sorting permutations by reversals and Eulerian cycle decompositions, SIAM Journal on Discrete Mathematics 12 (1999) 91–110.
[8] H. Kaplan, R. Shamir, R.E. Tarjan, Faster and simler algorithm for sorting sined permutations by reversals, in: Proc. 8th ACM–SIAM Symp. on Discrete Algorithms, 1997, pp. 344–351.
[9] G.A. Watterson, W.J. Ewens, T.E. Hall, A. Morgan, The chromosome inversion problem, Journal of Theoretical Biology 99 (1982) 1–7.
[10] A. Caprara, Formulations and complexity of multiple sorting by reversals, in: Proceedings of the 3th Annual International Conference on Computational Molecular Biology, 1999, pp. 84–93.
[11] I. Pe'er, R. Shamir, The median problems for breakpoints are NP-complete, Technical report TR98-071, The Electronic Colloquium of Computational Complexity, 1998.
[12] A. Solomon, P. Sutcliffe, R. Lister, Sorting circular permutations by reversal, in: Proceedings of the Workshop on Algorithms and Data Structures, Carleton Univ., Ottawa, Canada, Springer, 2003, pp. 1–10.
[13] D.A. Christie, R.W. Irving, Sorting strings by reversals and by transpositions, SIAM Journal on Discrete Mathematics 14 (2001) 193–206.
[14] A.J. Radcliffe, A.D. Scott, E.L. Wilmer, Reversals and transpositions over finite alphabets, Univ. of Cambridge, Isaac Newton Institute for Mathematical Sciences, IP02109-CMP.
[15] M. Hall Jr., The Theory of Groups, The Macmillan Company, New York, 1959.
[16] M.R. Jerrum, The complexity of finding minimum-length generator sequences, Theoretical Computer Science 36 (1985) 265–289.
[17] D.S. Hirschberg, The longest common subsequence problem, Ph.D. Thesis, Princeton University, 1975.
[18] D. Maier, The complexity of some problems on subsequences and supersequences, Journal of the ACM 25 (1978) 322–336.
[19] D.S. Hirschberg, Recent results on the complexity of common subsequence problems, in: D. Sankoff, J.B. Kruskal (Eds.), Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, Addison-Wesley, Reading, MA, 1983, pp. 325–330.
[20] R.W. Irving, C.B. Fraser, Two algorithms for the longest common subsequence of three (or more) strings, in: A. Apostolico, M. Crochemore, Z. Galil, U. Manber (Eds.), Proceedings of the Third Annual Symposium on Combinatorial Pattern Matching, in: Lecture Notes in Computer Science, vol. 644, Springer-Verlag, Berlin, 1992, pp. 214–229.
[21] H.L. Bodlaender, R.G. Downey, M.R. Fellows, H.T. Wareham, The parameterized complexity of sequence alignment and consensus, Theoretical Computer Science 147 (1995) 31–54.
[22] M. Frances, A. Litman, On covering problems of codes, Theory of Computer Systems 30 (1997) 113–119.
[23] J.K. Lanctot, M. Li, B. Ma, S. Wang, L. Zhang, Distinguishing string selection problems, in: Proc. 10th ACM-SIAM Symp. on Discrete Algorithms, 1999, pp. 633–642.
[24] D. Knuth, The art of computer programming, in: Sorting and Searching, vol. III, Addison-Wesley, Reading, MA, 1973.
[25] K.J. Räihä, E. Ukkonen, The shortest common supersequence problem over binary alphabet is NP-complete, Theoretical Computer Science 16 (1981) 187–198.
[26] B. Charlesworth, P. Sniegowski, W. Stephan, The evolutionary dynamics of repetitive DNA in eukaryotes, Nature 371 (1994) 215–220.
[27] M. Crochemore, M.-F. Sagot, Motifs in sequences: Localization and extraction, in: Handbook of Computational Chemistry, Marcel Dekker Inc., 2002.
[28] M.T. Hallet, An integrated complexity analysis of problems from computational biology, Ph.D. Thesis, Department of Computer Science, University of Victoria, Victoria, BC, Canada, 1996.
[29] V.G. Timkovskii, Complexity of common subsequence and supersequence problems and related problems, Kibernetika 5 (1989) 1–13. English Translation.